

Report On
A Framework for Sentiment Analysis: A Data-Driven
Approach

by

Md. Jahedul Islam
17101430

Tonmoy Sarker
17301052

Md. Shubiour Shuvo
17301132

Md. Robin Hossen
17301110

Minhaz Uddin Ahmedh
17301087

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
June 2021

© 2021. Brac University
All rights reserved.

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

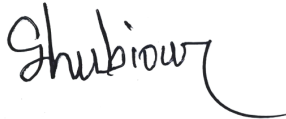
Student's Full Name & Signature:



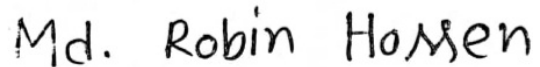
Md. Jahedul Islam
17101430



Tonmoy Sarker
17301052



Md. Shubiour Shuvo
17301132



Md. Robin Hossen
17301110



Minhaz Uddin Ahmed
17301087

Approval

The thesis titled “A Framework for Sentiment Analysis: A Data-Driven Approach” submitted by

1. Md. Jahedul Islam (17101430)
2. Tonmoy Sarker (17301052)
3. Md. Shubiour Shuvo (17301132)
4. Md. Robin Hossen (17301110)
5. Minhaz Uddin Ahmed (17301087)


Of Spring, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on June 2, 2021.

Examining Committee:

Supervisor:
(Member)

Mohammad Zavid Parvez
Assistant Professor
Department of Computer Science and Engineering
BRAC University

Co-supervisor:
(Member)

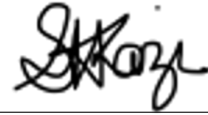


Md. Anisur Rahman
Academic
School of Computing and Mathematics
Charles Sturt University

Program Coordinator & Co-supervisor:
(Member)

Md. Golam Rabiul Alam
Assistant Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)



Sadia Hamid Kazi
Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Internet is free and straightforward access to an immense measure of crude content information that can be mined for sentiment analysis. For a long time, this is being used for market research, user opinion mining, recommendation systems, analyze people's views on a topic, etc. Many different techniques have been developed, yet a lot of complication remains. Selecting and understanding attribute patterns in a text dataset is important to build a good model and know where this model can be used. Different text datasets have different relations between their attributes and classes. For example, let's take a dataset with totally random English texts labelled as positive or negative. We expect to see that extracted attributes for the positive or negative class are very heavy with general words that we consider positive or negative in everyday English use. However, if the dataset is created on a niche topic, such as an economic, pandemic, etc, we would probably see that positive and negative classes are now heavy with words specific to these topics, or they may not be considered important at all by the classifier. However, we might want to give importance to those niche-specific attributes specifically. In this paper, we take five different datasets of different instance lengths. We use Weka as a tool and go through some attribute selection techniques, do sentence-level sentiment analysis, and finally extract patterns from the datasets to analyze them. There are few related works on these datasets and our technique performed better than the existing works. We have been successful to beat Fuzzy method in terms of accuracy and better extraction of polarity in texts. Our approach have been proven to better work with the datasets than many former methods. In this paper, we aim to present a method that can easily be fruitful to any dataset for text mining and can have a decent accuracy. In this paper, we aim to present a method that can easily be fruitful to any dataset for text mining and can have a decent accuracy.

Keywords: Sentiment Analysis, Attribute selection, Pattern Extraction, Classification, Accuracy, Application of Machine Learning.

Acknowledgement

We are thankful to Dr. Anisur Rahman, Academic in the School of Computing and Mathematics at Charles Sturt University, Australia. We are incredibly thankful and appreciative to him for his knowledge, and honest and important advise and support that he has provided to us.

Table of Contents

Declaration	1
Approval	2
Abstract	4
Acknowledgment	5
Table of Contents	6
List of Figures	7
List of Tables	8
Nomenclature	9
1 Introduction	10
2 Literature Review	11
3 Proposed Method	14
3.1 Data Collection	14
3.2 Data Preprocessing	16
3.3 Experimenting on the datasets	17
3.4 Finding acceptable combination for each dataset	18
3.5 Analyzing extracted patterns	18
4 Experimental result and discussion	19
4.1 Evaluation Criteria	19
4.2 Discussion on Finance phrase-bank (FN) dataset	19
4.3 Discussion on Stock Market (SM) dataset	20
4.4 Discussion on Sentiment140 (S140) dataset	22
4.5 Discussion on Movie Reviews (MR) dataset	23
4.6 Discussion on Climate Change (CC) dataset	24
4.7 Comparison	25
5 Conclusion	27
Bibliography	30

List of Figures

3.1	Flow chart for the proposed method	15
4.1	Line diagram of the FN dataset's accuracy across experiments	20
4.2	Line diagram of the SM dataset's accuracy across experiments	21
4.3	Line diagram of the S140 dataset's accuracy across experiments	22
4.4	Line diagram of the MR dataset's accuracy across experiments	24
4.5	Line diagram of the CC dataset's accuracy across experiments	25

List of Tables

3.1	A brief introduction on datasets	14
4.1	Comparison table with similar type work.	22
4.2	Comparison table with Fuzzy rule based work.	23
4.3	Accuracy of the Classifiers For the Best Technique.	25
4.4	Precision of the Classifiers For the Best Technique.	26
4.5	Recall of the Classifiers For the Best Technique.	26
4.6	F1 score of the Classifiers For the Best Technique.	26

Nomenclature

The next list describes several abbreviations that will be later used within the body of the document

CC Climate Change

FN Finance Phrase-bank

MR OMovie Reviews

S140 Sentiment140

SM Stock Market

Chapter 1

Introduction

Sentiment Analysis (SA) is a branch of analysis that integrates natural language processing (NLP), text analysis, computational linguistics, and biometrics in order to systematically identify, extract, measure, and analyse emotional states and subjective information [24]. A text analysis system SA uses a combination of natural language and machine learning approaches to give weighted feelings to the items, themes, and categories present inside a sentence or sentence. It helps data analysts in large organizations to determine public opinion, do thorough market research, monitor reputations of brands and products and understand user experiences. Moreover, data analytics organizations often include SA APIs from third parties in their customer experience management, social media monitoring and employee analytics platforms to provide their customers with significant insights.

In this time of age, analyzing the emotions of people in response to various events has been of great importance to understand and predict certain human behaviour patterns. The sudden emergence of urgent crises, for example, Covid-19 at the moment of writing, put a huge emotional toll on people. The Lockdown is having a dramatic impact on societies and economies around the world. People are constantly expressing their thoughts on social media in texts, which potentially carry their emotional information. This information could be valuable to make public decisions or understand how people's emotions change with time amid these types of circumstances. Through social media, people can freely communicate their ideas, opinions and feelings about different topical events, topics, etc. It is a wonderful technique to collect information about the opinions of individuals on any topic. With the emergence of Web 2.0, social media messages are becoming more informational since they contain, as opposed to conventional text-only posts, visual content as well as text. The aim of sentiment analysis is to determine the underlying features of the posts [21].

Chapter 2

Literature Review

Depending on their application, social media is of four types- Content communities (Youtube, Instagram), Social networking (Facebook, LinkedIn), Blogs (Reddit, Quora), and Micro-blogs (Twitter, Tumblr). Among them, Twitter is the most popular media platform for collecting user opinions [19]. Twitter, in particular, is a public domain where anyone can see any tweet without permission. In a study, researchers have used this opportunity to understand people's reactions to global issues like climate change and analyze them [27]. They have used word clouds and figured out the frequency of words used in a sentence to summarize the entire content. While data preprocessing, researchers excluded useless tweets to optimize the data and to make it more relevant to the study [27][10]. For instance, monosyllable tweets that have no meaning are removed, and posts representing complicated topics have been excluded. However, In another paper [19] researchers recommended applying lexicon-based for small datasets, in our case on the small dataset we have applied machine learning-based approaches and gained quite pleasant accuracy [28], [29]. In a recent paper, the fuzzy rule-based approach was demonstrated to deal with multimodal sentiment analysis.

It can calculate feelings with multiple sentiment classifications for datasets. Typically, two-class datasets have a purely positive feeling, while three-class datasets have a neutral feeling label[23][17]. Strong and famous tools for pattern detection and classification are categorization systems based on fuzzy rules. These systems are particularly efficient to cope with uncertainty, ambiguity or vagueness thanks to their flexibility. Many scholars have used machine learning methods such as Naive Bayes and SVM to analyze tweet sentiment. Its methodology is automated and consists of three basic phases: preprocessing text, development of sentiment lexicons, and categorization of emotion polarity utilizing a fuzzy rule framework. As they tried to demonstrate, the fuzzy rule process takes less time to bring out the result, but their precision and recall scores are way lower than ours. There is related work on sentiment140 dataset which is one of the datasets we used in this paper[23].

Another work on social media mining for product modelling utilized a model based on the Latent Dirichlet Allocation (LDA), noted for its superior performance between multiple topic modelling algorithms in large-scale documentation, interpreting the latent topics identified[20]. The lexicon approach employs the predefined dictionaries that define the terms of sentiment and their accompanying feeling (e.g.

SentiWordNet) and detects the sentimental approach of a text-based on the semanticized orientation of the words or phrases in document[23][9][22][15].

HEMOS (humor-EMOji-Slang) has done a great job of classifying fine-grained feelings in China with a deep learning technique. We evaluated the important aspect of recognizing the effect on the job of affective media processing of humour, pictograms and slang[26][11].

In another paper, we found that they used a novel meta-heuristic method(CSK). And this method depends on the K- means and cuckoo search. They tried to find the optimum cluster-heads from a sentimental feature of the Twitter dataset. They also compared their method with an SVM tree and a Naive Bayes tree. However, they struggled to deal with sarcasm and irony tweet [18].

Most of the research work done on sentiment analysis has focused primarily on contexts of a sentence to gain knowledge and understand patterns. The authors of this paper [14] have instead focused on aspect level sentiment analysis to get more complete and in-depth results. Context varies on different aspects which makes a wide impact in understanding the whole sentiment of a sentence. The polarity of sentiment depends not only on context but also on aspects. The authors have implemented attention-based LSTM to classify aspect level sentiments. In the paper, the 3 main labels used in LSTM are positive, negative, neutral. As for different aspects, we can get different outcomes, authors have proposed to learn aspect based word embedding. Standard LSTM has not used in this paper rather a different attention mechanism to gather key information's from a sentence through aspect level mining. To take more advantage of aspect information, input aspects are being embedded into word vectors. The dimensions expand if the aspect embeddings are added to LSTM. Additional parameters such as AT LSTM, AE LSTM, ATAE LSTM are also used by the authors. The dataset used in this experiment is derived from SemEval 2014 Task 4 2 comprising customer reviews differing based on different aspects. The comparison of this proposed method with the baseline model shows the difference between LSTM, TD LSTM and AE LSTM, ATAE LSTM. The result shows that the proposed method gives slightly more accuracy in 3 class prediction than standard LSTM. In conclusion, we can say that the proposed method gives us a more in-depth analysis of sentiment than most other context-based sentiment analysis. The key point here to use the aspect embeddings and make use of aspect information that can give a clear understanding of the polarity of a sentence. The potentials and opportunities of this method are proven through the superior performance of AE LSTM and ATAE LSTM.

Sentiment analysis was performed on a review dataset by some researchers [13]. Data mining techniques are used to extract critical information from large databases and classify it as positive or negative. The information gathered through comments, reviews, and critiques is classed according to its polarity, which indicates whether the information is positive or negative.

They discovered that Naive Bayes performed better for film reviews than K-NN did for hotel reviews. Between Naive Bayes and K-NN, there is a distinction in terms of the training dataset utilized in the data preparation stage for K-NN. To get the

score for each word in the training dataset, the chi-squared test is utilized. They evaluated the sentiment analysis's performance using the accuracy, precision, and recall criteria. The term "accuracy" refers to the general correctness of various sentiment models. When the number of reviews for a film is around 4500, Naive Bayes is 82.43 percent accurate, whereas K-NT is 69.81 percent accurate. In hotel reviews, Naive Bayes has a 55.09 percent accuracy rate, whereas K Nines has a 52.14 percent accuracy rate. If they used alternative data cleaning and preprocessing techniques, the accuracy would be increased.

Adequate levels of soil fertility are necessary to achieve and maintain if agricultural land is to remain productive. There was also research on soil fertility dataset using J48 classification techniques [6]. They used three techniques (NBTree, SimpleCart, J48) with CfsSubsetEval attribute selection, where J48 turned out to be the best classifier. This paper's drawback is that they used only 2000 instances of the dataset, which is a very small dataset. It's hard to tell that it will give the exact same result for a larger dataset with the same approach. Still, we demonstrated in our paper using different classifiers along with J48 for 5 different datasets [1]–[3]. However, in our case, CfsSubsetEval performed worst for most datasets.

For our paper, we collected five datasets and used a few different classifiers to justify the results between experiments fairly [28], [29]. Classification and prediction are two types of data analysis that can be utilized to extricate models portraying significant information classes or anticipating future information patterns. Classification is a method of data mining used to discover which data instances belong to which group. We first tested all those using different classifiers on our datasets to see the result with different amounts of attributes, using other ranking methods such as Info gain with ranker, CfsSubset with best first. Our goal was to increase the accuracy up to 7-8% by reducing the attributes from datasets and to compare the result among all those classifiers and how they are reacting in a different pattern. We showed all our experimental results in a graph to have a better visualization with the explanation. We also demonstrated a better result and accuracy in our experiment result section, which is relatively easier than a Fuzzy rule-based approach [23]. Moreover, compared to the existing relevant techniques, our proposed method is less complicated to apply to any text datasets. The contributions of the paper are as follows:

- Our method is easy to use for a dataset compared to the existing methods,
- Performance of our method is better than some existing works.

The rest of the essay is constructed accordingly. In Section II, we described our recommended method. We detailed our strategy in Section III and displayed experimental results. We explored the conclusion of the paper and the next work in Section IV.

Chapter 3

Proposed Method

In this section, we discuss our method for extracting patterns and analyzing different types of sentiment labelled datasets. There are five steps to that,

- Data collection
- Data preprocessing
- Experimenting on the datasets
- Finding an acceptable accuracy for each dataset
- Analyzing extracted features

Figure-1 portrays the flow chart for the above-mentioned steps.

3.1 Data Collection

We have collected five different datasets of different types and lengths (Table 1). All the sources for the datasets are mentioned on their respective description below.

Table 3.1: A brief introduction on datasets

Dataset	Records	Class
Finance phrase-bank (FN)	4802	Positive=1347, Neutral=2857, Negative= 598
Stock Market (SM)	5761	Positive=3669, Negative=2092
Sentiment140 (S140)	21411	Positive=12486, Negative=8925
Movie Reviews (MR)	9916	Positive=4992, Negative=4924
Climate Change (CC)	3455	Yes=2431, No=1024

FN dataset is a human-annotated finance phrase-bank [7]. First, relevant news headlines were collected from multiple sources. The collected texts were annotated by 16

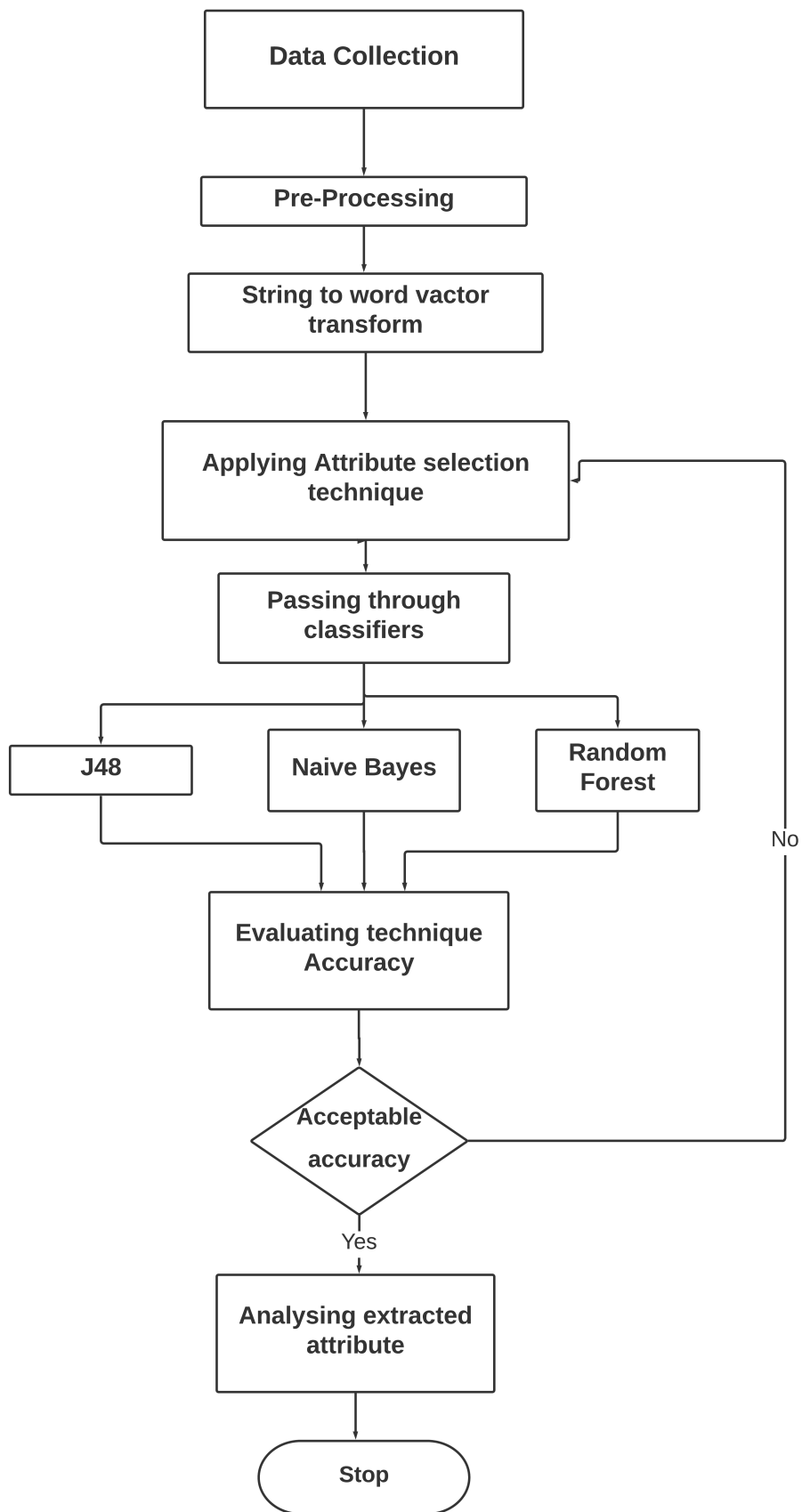


Figure 3.1: Flow chart for the proposed method

people with adequate background knowledge on financial markets. Three of the annotators were researchers, and the remaining 13 annotators were master’s students at Aalto University School of Business with majors primarily in finance, accounting, and economics. All of the texts were annotated either positive, negative, or neutral. SM dataset consists of texts gathered using multiple Twitter handles on the topic of the Stock Market news [25]. Collected texts were then manually labeled positive or negative in the context of the stock market.

S140 is sentiment140 [4] dataset, creator of this dataset automatically gathered tweets with the help of Twitter Search API by using a keyword search. Unlike most sentiment datasets here, instead of manually labeling tweets by humans, tweets with positive emoticons like :) were assumed positive and ones with negative emoticons like :(were assumed negative. This dataset initially consisted of 1.6 million tweets of totally random topics. As working with such a large dataset would be hard and time-consuming, we filtered and selected only tweets that were posted on April 18, 2009. We still end up with quite a large dataset but decided to work on it.

MR dataset contains movie reviews and their associated binary sentiment polarity labels [5]. The core dataset contains 50,000 reviews split evenly into 25k train and 25k test sets. The overall distribution of labels is balanced (25k pos and 25k neg). In the entire collection, no more than 30 reviews are allowed for any given movie because reviews for the same movie tend to have correlated ratings. In the labeled train/test sets, a negative review has a score less than or equal to 4 out of 10, and a positive review has a score greater or equal to 7 out of 10. Thus reviews with more neutral ratings are not included in the train/test sets. Similar to sentiment140, we worked on a reduced version of this dataset. We merged train, and test sets, shuffled them, and randomly selected 10,000 instances, 5000 from positive and 5000 negative.

CC dataset consists of tweets on the topic of Climate Change [12]. Contributors evaluated tweets for belief in the existence of global warming or climate change. The possible answers were "Yes" if the tweet suggests global warming is occurring, "No" if the tweet suggests global warming is not occurring, and "I can’t tell" if the tweet is ambiguous or unrelated to global warming. Because of the ambiguity of the "I can’t tell" class, we dropped it from the dataset and only worked with the remaining two classes.

3.2 Data Preprocessing

We more or less performed the same preprocessing tasks for all the datasets. The only exception being the removal of stopwords and stemming, more on it will be explained below. We first started by cleaning up all the datasets. Removed any links, user tags, hashtags, numbers, punctuations, and special characters and lowercase transformed all the text. A hand-coded python script was created for this purpose. That way, it was easy to handle what actions we take on the string contents. Multiple lines of regular expression substitution rules were ordered in way that do not strip too much semantic value from the texts.

After our initial clean up, we loaded our datasets onto Weka software. For this paper, we used Weka as our main tool [3]. Weka has nicely created filters that allow

users to perform further preprocessing tasks with ease. It lets us visualize selected attributes very thoroughly after the word vector creation. Before creating the word vectors, we used RemoveDuplicates filters to remove all the duplicate instances. After that, we used the StringToWordVector filter to create word vectors from the datasets. While applying the filter, there are multiple options we can choose. We can decide if we want to consider tf-idf (term frequency-inverse document frequency), a numeric measure representing how relevant a word is to a document or corpus. Stemming algorithms, which strip words into their root form, here three different options are available to choose from- IterativeLovinsStemmer, LovinsStemmer, and SnowballStemmer. Or we can choose not to perform stemming at all. For the stopwords list, there are MultiStopwords, Rainbow, etc. As for the tokenizer, we kept it as default, which is WordTokenizer. In addition to all these, we also have the liberty to choose how many words should the filter try to keep for each class. For experiments, we will be rotating around tf-idf, stemmers, stopwords, and how many attributes initially we try to keep per class during word vector creation. Before passing them onto the classifiers, we transformed all numeric attributes to nominal attributes for our benefit using the NumericToNominal filter.

3.3 Experimenting on the datasets

As stated above, we will be selecting words/attributes during word vector creation by changing what we consider or apply during the process. We will also be testing with two attribute selection filters among several that are available in Weka- CfsSubsetEval with best-first search method and InfoGainAttributeEval with ranker search method. CfsSubsetEval evaluates the worth of a subset of attributes by considering each feature's predictive ability and the degree of redundancy between them. In contrast, InfoGainAttributeEval evaluates the worth of an attribute by measuring the information gain with respect to the class. To get a fair justification of our experiments, we choose Naïve Bayes, Random Forest, J48 as our classifiers. The idea is to try out different combinations of choices and see which gives acceptable accuracy on all of the classifiers. We initially wanted to include SVM(Support Vector Machine classifier), Multilayer Perceptron, and along with the three mentioned classifiers to get a more accurate view. However, after running some experiments, we saw they were taking too much time to conclude, and it would become very time-consuming for us to continue working with them across all five datasets. So, we ended up working with only three.

We ran our first classification test parallelly on all datasets with the word vector created by keeping default values for the StringToWordVector filter. No stemming was done, stopwords were not removed, and tf-idf was not considered, and how many words per class should try to keep was set to 1000. For five datasets, their base accuracies were documented as such. Now, we gradually try to increase accuracy from there. We then try to see if accuracy increases if tf-idf is considered during vector creation or does it decrease accuracy. We observe what accuracy we get if we use different stemming algorithms, what accuracy we get after removing stopwords, do stemming overall decreases accuracy, or increase initially but decrease if stopwords were removed along with them. We also increase the per class attribute count to see if accuracy will increase given more words are taken into consideration. While increasing initial per class attribute counts, we also test among two attribute

selection methods to see which one selects better attributes from the word vector created on the current combination.

3.4 Finding acceptable combination for each dataset

After running multiple experiments, we stop when we see overall accuracies decreasing. We compare our results and select the only experiment where all the classifiers' accuracy is on the range of acceptability. Even if any classifier gave a relatively high performance in other experiments than the chosen one, the other two classifiers might be giving poor results. We then try to improve the accuracy of J48 by adjusting some hyperparameters, while the combination is the same as the accepted one. J48 generates a tree of selected attributes that lets us visually analyze them better.

3.5 Analyzing extracted patterns

After obtaining the tree, we observe how relevant each attribute is with each other or how unrelated yet closer in the tree. We can determine these obtained attributes only niche to this particular dataset, or can we use them to build a model for general sentiment analysis of datasets on the same topic. This extracted pattern can be used with any other popular sentiment classifier algorithm such as Support Vector Machine to get a more accurate and better model. We can also tell if a certain type of dataset is not best suited to the word vector approach.

Chapter 4

Experimental result and discussion

For the first experiment across all datasets, we did not consider tf-idf, no stemming was done, stopwords were kept as it is, and for each class, 1000 words were attempted to keep. Setting that as our base, we started our experimentation. We tried every single available stopwords list separately and found the Rainbow stopwords list to be the only one that contributes to improving most datasets' accuracy. We did not test with any custom stopwords list, as we would have had to create five different lists for five datasets. Next, we tested all the available stemming algorithms, while word vector was created attempting to keep 1000 words (words to keep 1000) per class. Among IterativeLovinsStemmer, LovinsStemmer, and SnowballStemmer, IterativeLovinsStemmer was giving some good results, but after going through the selected attribute list, we saw many noise attributes were added to the list. So, we decided not to use any stemmer algorithm at all. Continuing with tf-idf, words to keep, different attribute selection filter combinations and their outcomes for each dataset will be explained below. The performance of our method is evaluated in terms of accuracy, precision, recall, and F1-score [8].

4.1 Evaluation Criteria

Before going into details of the experiments, we want to clarify how we are deciding which technique is performing better. We measure accuracy on three different classifiers and compare the overall accuracy of different experiments to see where it is highest. An easy way to find out the overall accuracy increase or decrease is to compute all the classifiers' average accuracy.

$$Avg. acc. = \frac{NB acc. + RF acc. + J48 acc.}{3}$$

4.2 Discussion on Finance phrase-bank (FN) dataset

Financial-Phrase bank dataset consists of positive and negative words that are mostly niche to finance topics. Hence, stemming might strip all of them to general form, so we did not perform any stemming on this dataset. On our initial base test for this dataset, 1076 attributes were selected and without dropping any of the attributes before classification, we obtained an average of 73.27%. Next, removed

stopwords with Rainbow list and applied CfsSubsetEval attribute evaluator on the word vector of 1076 attributes, and the selector reduced the list to 41, and the average accuracy decreases to 71.84%. Continuing with our experiment, we then recreated the word vector again. This time per class, we tried to keep 1500 words and ended up with an initial attributes count of 1640. Without any attribute selection, just by removing stopwords, we get an average of 73.33%, a slight increase than before. After applying CfsSubsetEval on the vector, it selects 68 attributes, and the classifiers' average accuracy drops to 71.46%. We can see that CfsSubsetEval is not showing any good results. We consider tf-idf during vector creation and obtain 1640 attributes again; we then apply InfoGainAttributeEval with the ranker search method. We specify that we want to select 100 attributes. From 101 selected attributes, we obtain a score of 73.22%. We do the same thing again, except this time without considering tf-idf, and the average raises to 73.55%. We also experiment to see if we do not remove stopwords on the previous settings what happens to the score, and it only drops by 0.04%. So, technically this dataset gives surprisingly good accuracy if words are not stripped too much of their semantic meanings. Additionally, we tried to keep 3000 words per class, where we obtain a word vector of 4236 attributes, which gives a score of 69.64%, which is very poor but expected. And applying InfoGainAttributeEval on the same vector gives 73.22%. So comparing all we see that for this dataset when we try to keep 1500 words per class, also use Rainbow stopwords list for stopwords removal while creating word vector, and then apply InfoGainAttributeEval to select attributes, we get the best average accuracy score of 73.55%.

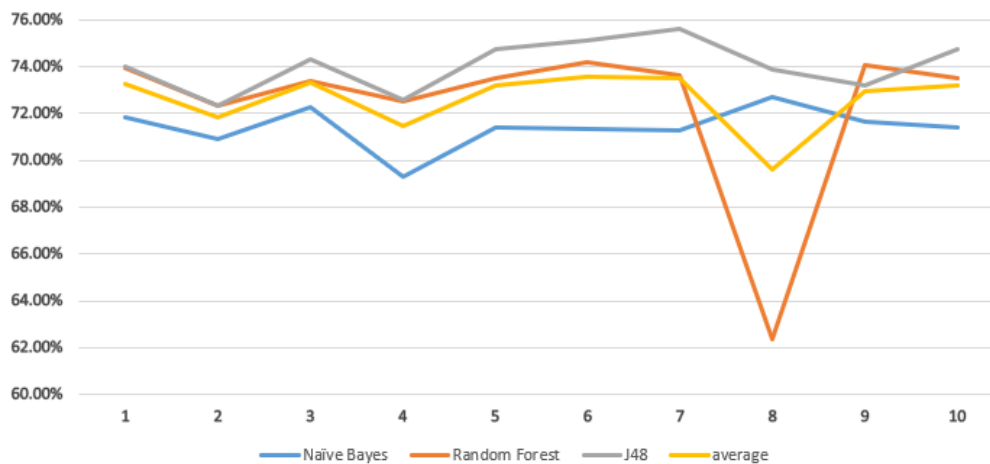


Figure 4.1: Line diagram of the FN dataset's accuracy across experiments

4.3 Discussion on Stock Market (SM) dataset

For this dataset, we got an average base score of 77.86%. This dataset is already giving really good accuracy results. After that, we test how tf-idf affects our dataset. We set words to keep to 1000 similar to our base experiment, but we removed stopwords using rainbow (stopwords) and we considered tf-idf when creating the word vector. The average accuracy drops to 77.40%. Next, we do not consider

tf-idf, everything same as before, except after creating the word vector, we use CfsSubnetEval to select attributes. From 1602 initially selected attributes, this evaluator selects 126 attributes and the average drops by 2%. Again, the same as before, only InfoGainAttributeEval is used and it selects 101 attributes from 1602 and gives an average accuracy of 76.39%. Furthermore, we test CfsSubsetEval and InfoGainAttributeEval evaluators with tf-idf considered, Rainbow stopwords, words to keep 1000, and we get average accuracy of 76.10% and 76.25% respectively. Because accuracies between classifiers for this dataset is already really good. We directly jumped to work with very large attribute counts. We choose 3000 words to keep per class while creating word vectors, where tf-idf is not considered and Rainbow stopwords used. Our created word vector consists of 7931 attributes now. Instead of trying to work with all of them, we apply InfoGainAttributeEval to select the first 300, then 400, 500, 600, and so on up to 800 attributes. We measure for every selection and starting with 78.57% for 300, 78.98% for 400, 79.12% for 500, and the average accuracy increases till 800 becoming 79.52%. And overall accuracy score still keeps increasing for a while, with us increasing the number of attributes we select.

So, for this dataset, we can see similar to finance-phrase bank InfoGainAttributeEval selector performed quite well. Unlike the previous dataset here, we had to create a word vector with a larger number of attributes initially and then select them to obtain maximum performances.

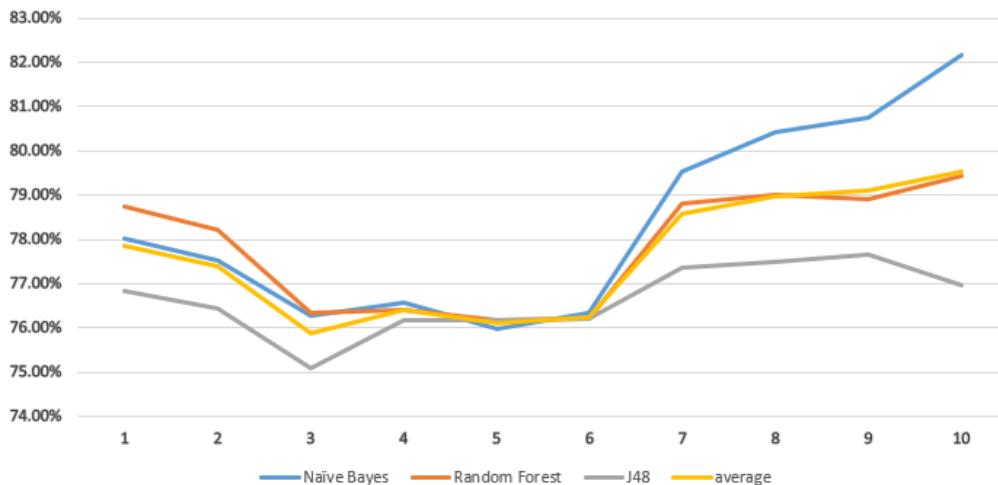


Figure 4.2: Line diagram of the SM dataset's accuracy across experiments

Researchers devised a novel method of stock market research [16] that included a method for determining the buy or sell signal to investors. By using both stock-related RSS news feed and the Sensex points, they claim that they would be able to predict the market using the suggested method. With the use of J48, Naive Bayes, and Random Forest classifiers, we were able to get better accuracy with our dataset.

Table 4.1: Comparison table with similar type work.

RSS news feed	ID3	C4.5	Moving Average	Moving Average+ Sentiment Analysis
Precision %	46.69%	47.49%	64.32%	78.75%

Our works	J48	Naive Bayes	Random Forest
Precision %	76.9%	82.0%	79.2%

4.4 Discussion on Sentiment140 (S140) dataset

Similar to the base experiment our our second test, the accuracy stays the same if we only consider the tf-idf and where the initial attribute count is 1078. On our next experiment, the accuracy goes slightly higher giving an average score of 71.76%, where we applied InfoGainAttributeEval and select 101 attributes from the initial 1078. However, when we applied CfsSubsetEval in the next experiment, the Naive Bayes and J48 accuracy fall down to 68%, dropping down the average to 69.91%. Continuing with two consecutive tests, we got a similar accuracy of 69%, where we are not selecting any attributes and only testing with tf-idf. After that, we tried to keep 3000 words per class, removed stopwords and did not consider tf-idf while creating word vector, we obtain an initial count of 3530 attributes, we then apply InfoGainAttributeEval to select 300 attributes. We get our peak average accuracy score of 73.53% from 301 selected attributes for this dataset. On the contrary, when we tested both attribute selectors InfoGainAttributeEval and CfsSubsetEval to choose 101 and 73 attributes respectively from the initial attributes count of 1638 (words to keep 1500) while tf-idf was considered. The average accuracy never goes over 71%. Hence, we select 73.53% to be our peak average and adjacent technique to be the best for the S140 dataset.

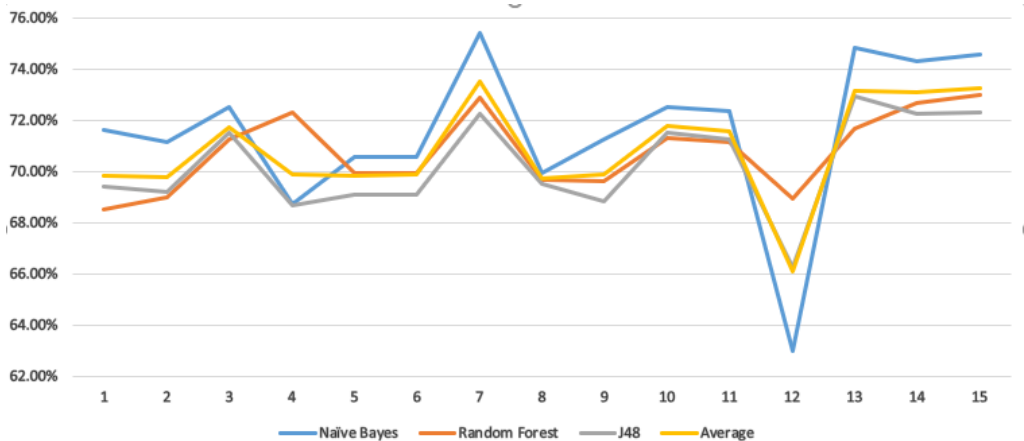


Figure 4.3: Line diagram of the S140 dataset's accuracy across experiments

Furthermore, we were able to achieve higher Accuracy, F1, precision, and recall score than the fuzzy rule-based technique for the exact same dataset. The precision, Recall, F1 score differences with their work which is higher than them is shown below [23].

Table 4.2: Comparison table with Fuzzy rule based work.

Dataset(S140)	Precision	Recall	F1 Score
Fuzzy Rule Based	62.8%	66.1%	77.2%
Ours Work	74.3%	73.9%	52.8%

4.5 Discussion on Movie Reviews (MR) dataset

For this dataset we obtained a base average accuracy of 79.53%, which is really good. We almost wanted to accept it as it is, but we still had to experiment further to better understand the dataset. We also tested and saw Rainbow stopwords list works quite well with this dataset and so we applied it for rest of the experiments. On our next experiment, we try to keep 1000 words per class and consider tf-idf while creating the word vector. Initially it selects 1160 attributes, we then applied CfsSubsetEval, which selected 54 attributes from 1160. After passing the obtained word vector through classifiers we see that our average accuracy drops to 76.82%. We then do the same test except we do not apply any attribute selector this time, and our accuracy score raises to 80% for both Naive Bayes and Random Forest, however the J48 goes to 72% and we obtain an average score of 78.65%. Accuracy stays almost same for next three consecutive experiments. In the next experiment, we again apply CfsSubsetEval on a word vector initially consisted of 2861 attributes, and the evaluator reduces the list to 60. Here, we tried to keep 2500 per class and considered tf-idf. After running classification on the word vector with now 60 attributes, we see higher accuracy scores for all three classifiers and the average being 77.69%. Then in our next experiment, we see an accuracy drop if we do not select any attributes for the same case. Finally, unlike previous datasets here very high average accuracy did not ensured that all classifier accuracies were in an acceptable range. So, even being slightly lower than the highest average accuracy, we choose the experiment with average score of 77.69%, where tf-idf was considered, attributes were selected using CfsSubsetEval. Also unlike other datasets this one performed well with CfsSubsetEval and poorly with InfoGainAttributeEval.

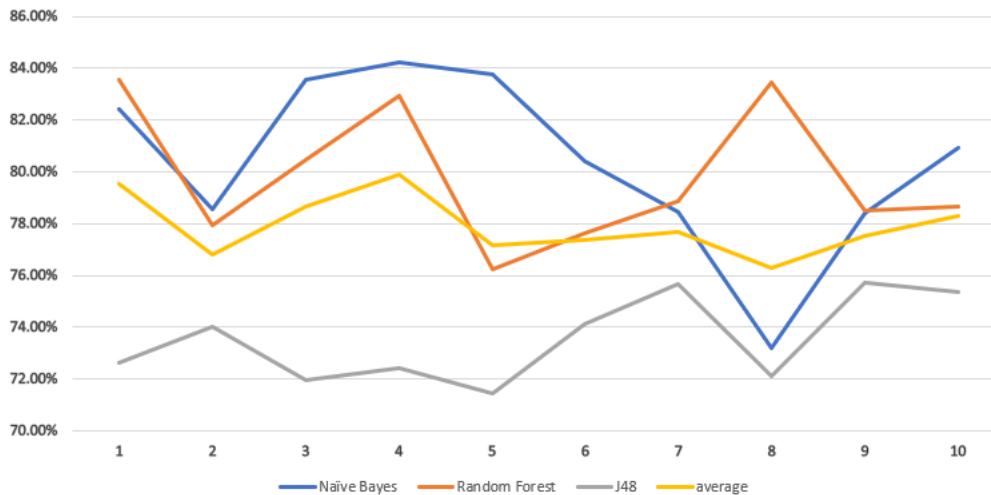


Figure 4.4: Line diagram of the MR dataset’s accuracy across experiments

4.6 Discussion on Climate Change (CC) dataset

In the base experiment we obtain 79.31% for Naive Bayes, 77.25% for Random Forest and 76.29% for J48 and the average accuracy is 77.62%. On the next experiment, Naive Bayes and J48’s accuracy falls to 74%, the Random Forest’s accuracy goes up to 78% and average accuracy drops to 75.30%, here we considered tf-idf while creating word vector. We then selected 73 attributes from the initial attribute count of 1587 by applying CfsSubsetEval. We saw Naive Bayes accuracy dramatically peaks at 79% . J48 in 75% and average accuracy being 75.30% in the experiment after that. However, this time the Random Forest accuracy score decreases by 2% from the previous experiment. Here we did not select any attribute and tf-idf was not considered. In next three consecutive experiments, the Naive Bayes’s accuracy again falls between 73-74% and J48, and the Random forest’s accuracy slightly stables at 74-78% and the average accuracy in a range of 74.87-75.30%. Finally, on the next experiment we got a stable and highest accuracy peak for all the three classifier, accuracies being in the range of 77-76% and the average accuracy being 77.19%. For this experiment did not apply any attribute selector and also tf-idf was considered during word vector creation. However, in the next two experiment after that, the J48 and Naive Bayes’s accuracy fall to 73% and the average accuracy being close to 75.26%, where we selected 73 and 74 attributes from the initial attribute of 5084 and 5083 respectively by applying InfoGainAttributeEval and CfsSubsetEval. Therefore, we choose the experiment with average accuracy score of 77.19% to be our best score and select the related method to be the best method for this dataset. Where, we tried to keep 3000 words per class, considered tf-idf during word vector creation and we did not selected any attributes applying attribute evaluators.

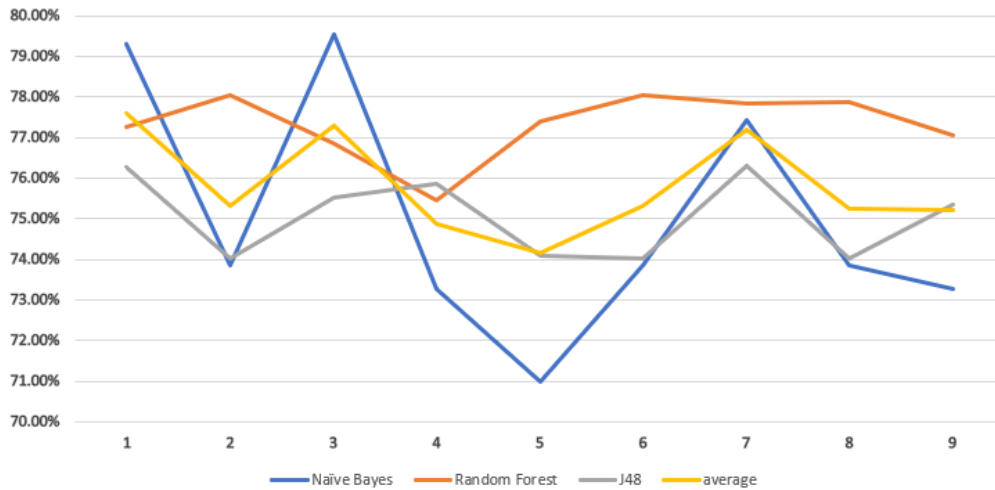


Figure 4.5: Line diagram of the CC dataset’s accuracy across experiments

4.7 Comparison

We have compared our method of analyzing sentiment based on different datasets with the Fuzzy rule method [23] and a new method [16] that combines both RSS feeds and Sensex points. We have found our method to give better results on specific datasets in comparison to Fuzzy rule. As shown in Table 4.2 we can see that regarding the S140 dataset we find that our data driven approach gives 74.3% precision compared to 63% precision as well as 74% recall accuracy where the Fuzzy rule gives only 66% recall accuracy. Although we have a significantly less F1 score than Fuzzy rule. We can admit that for other 3 datasets (CC, FN, MR) we can find similar result comparisons on which our method will outweigh the performance.

Table 4.3: Accuracy of the Classifiers For the Best Technique.

Dataset	J48	Naive Bayes	Random Forest
FN	75.12%	71.36%	74.19%
SM	76.98%	82.16%	79.43%
S140	72.05%	73.98%	71.85%
MR	78.42%	78.49%	75.73%
CC	77.83%	76.32%	76.32%

Table 4.4: Precision of the Classifiers For the Best Technique.

Dataset	J48	Naive Bayes	Random Forest
FN	74.8%	70.4%	73.6%
SM	76.9%	82.0%	79.2%
S140	72.7%	74.3%	72.4%
MR	72.6%	82.5%	83.7%
CC	74.9%	79.9%	76.4%

Table 4.5: Recall of the Classifiers For the Best Technique.

Dataset	J48	Naive Bayes	Random Forest
FN	75.1%	71.4%	74.2%
SM	77.0%	82.2%	79.8%
S140	72.1%	73.9%	71.9%
MR	72.6%	82.4%	83.6%
CC	76.3%	79.3%	77.3%

Table 4.6: F1 score of the Classifiers For the Best Technique.

Dataset	J48	Naive Bayes	Random Forest
FN	73.6%	69.4%	72.4%
SM	75.8%	81.8%	78.8%
S140	70.2%	72.6%	70.1%
MR	72.6%	82.4%	83.6%
CC	74.5%	79.5%	74.7%

Chapter 5

Conclusion

In this paper, we presented a method for sentiment analysis from text data using various data mining and machine learning techniques. We used five different datasets of different instance length, performed sentence-level sentiment analysis, and analyzed via extracting patterns from those five datasets. We are able to secure almost 76-80% accuracy using the three classifiers for those datasets. Furthermore, we have tried to apply MulilayerPerception (neural network), but it takes a larger computation time because of the nature of the data then, while preprocessing we have not considered any symbol and emoticons which can hold a vast sentiment in a text. This is just a generalized framework that can be applied in all kinds of datasets for sentiment analysis and have decent accuracy, and it can be improved a lot by doing a bit more research. In the future, we are planning to use more classifiers along with some preprocessing techniques that can assure a higher accuracy. Moreover, we will try to visualize the data through PCA and tSNE.

Bibliography

- [1] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993, ISBN: 1558602380.
- [2] J. R. Quinlan, “Improved use of continuous attributes in c4.5,” *J. Artif. Int. Res.*, vol. 4, no. 1, pp. 77–90, Mar. 1996, ISSN: 1076-9757.
- [3] E. Frank, M. A. Hall, R. Holmes G. and <https://www.overleaf.com/project/60b4f8ab7a456a230e> Kirkby, B. Pfahringer, and I. H. Witten, “Weka: A machine learning workbench for data mining,” in *Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*, O. Maimon and L. Rokach, Eds. Berlin: Springer, 2005, pp. 1305–1314.
- [4] A. Go, R. Bhayani, and L. Huang, “Twitter sentiment classification using distant supervision,” *CS224N Project Report, Stanford*, vol. 1, no. 12, p. 2009, 2009.
- [5] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning word vectors for sentiment analysis,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Jun. 2011, pp. 142–150.
- [6] J. Gholap, “Performance tuning of j48 algorithm for prediction of soil fertility,” *ArXiv*, vol. abs/1208.3943, 2012.
- [7] P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala, “Good debt or bad debt: Detecting semantic orientations in economic texts,” *Journal of the Association for Information Science and Technology*, vol. 65, no. 4, pp. 782–796, 2014.
- [8] M. A. Rahman, “Automatic selection of high quality initial seeds for generating high quality clusters without requiring any user inputs,” *PhD Desertification, School of Computing and Mathematics, Charles Sturt University*, 2014.
- [9] E. M. Cody, A. J. Reagan, L. Mitchell, P. S. Dodds, and C. M. Danforth, “Climate change sentiment on twitter: An unsolicited public opinion poll,” *PLOS ONE*, vol. 10, pp. 1–18, Aug. 2015.
- [10] A. P. Kirilenko, T. Molodtsova, and S. O. Stepchenkova, “People as sensors: Mass media and local temperature influence climate change discussion on twitter,” *Global Environmental Change*, vol. 30, pp. 92–100, 2015.
- [11] M. Biltawi, W. Etaiwi, S. Tedmori, A. Hudaib, and A. Awajan, “Sentiment classification techniques for arabic language: A survey,” in *2016 7th International Conference on Information and Communication Systems (ICICS)*, 2016, pp. 339–346.

- [12] CROWDFLOWER. (2016). “Sentiment of climate change,” [Online]. Available: <https://data.world/crowdfLOWER/sentiment-of-climate-change>.
- [13] L. Dey, S. Chakraborty, A. Biswas, B. Bose, and S. Tiwari, “Sentiment analysis of review datasets using naïve bayes‘ and k-nn classifier,” *International Journal of Information Engineering and Electronic Business*, vol. 8, pp. 54–62, Jul. 2016. DOI: 10.5815/ijieeb.2016.04.07.
- [14] Y. Wang, M. Huang, X. Zhu, and L. Zhao, “Attention-based lstm for aspect-level sentiment classification,” Jan. 2016, pp. 606–615. DOI: 10.18653/v1/D16-1058.
- [15] K. Z. Aung and N. N. Myo, “Sentiment analysis of students’ comment using lexicon based approach,” in *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, 2017, pp. 149–154.
- [16] S. Bharathi.Sv and A. Geetha, “Sentiment analysis for effective stock market prediction,” *International Journal of Intelligent Engineering and Systems*, vol. 10, pp. 146–154, Jun. 2017. DOI: 10.22266/ijies2017.0630.16.
- [17] H. Liu and E. Haig, “Fuzzy rule based systems for interpretable sentiment analysis,” Feb. 2017.
- [18] A. Pandey, D. Rajpoot, and M. Saraswat, “Twitter sentiment analysis using hybrid cuckoo search method,” *Information Processing Management*, vol. 53, pp. 764–779, Jul. 2017.
- [19] Z. Drus and H. Khalid, “Sentiment analysis in social media and its application: Systematic literature review,” *Procedia Computer Science*, vol. 161, pp. 707–714, 2019.
- [20] B. Jeong, J. Yoon, and J.-M. Lee, “Social media mining for product planning: A product opportunity mining approach based on topic modeling and sentiment analysis,” *International Journal of Information Management*, vol. 48, pp. 280–290, 2019.
- [21] Y. Shi, L. Zhu, W. Li, K. Guo, and Y. Zheng, “Survey on classic and latest textual sentiment analysis articles and techniques,” *International Journal of Information Technology & Decision Making*, vol. 18, no. 04, pp. 1243–1287, 2019.
- [22] S. Taj, B. B. Shaikh, and A. Fatemah Meghji, “Sentiment analysis of news articles: A lexicon based approach,” in *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2019, pp. 1–5.
- [23] S. Vashishtha and S. Susan, “Fuzzy rule based unsupervised sentiment analysis from social media posts,” *Expert Systems with Applications*, vol. 138, p. 112 834, 2019.
- [24] Z. Zhao, H. Zhu, Z. Xue, Z. Liu, J. Tian, M. C. H. Chua, and M. Liu, “An image-text consistency driven multimodal sentiment analysis approach for social media,” *Information Processing Management*, vol. 56, no. 6, p. 102 097, 2019.
- [25] Y. Chaudhary. (2020). “Stock-market sentiment dataset, version 1,” [Online]. Available: <https://www.kaggle.com/yash612/stockmarket-sentiment-dataset/version/1>.

- [26] D. Li, R. Rzepka, M. Ptaszynski, and K. Araki, “Hemos: A novel deep learning-based fine-grained humor detecting method for sentiment analysis of social media,” *Information Processing Management*, vol. 57, no. 6, p. 102 290, 2020.
- [27] M. L. Loureiro and M. Alló, “Sensing climate change and energy issues: Sentiment and emotion analysis with social media in the u.k. and spain,” *Energy Policy*, vol. 143, p. 111 490, 2020.
- [28] M. A. Rahman, Y. Al-Saggaf, and T. Zia, “A data mining framework to predict cyber attack for cyber security,” in *2020 15th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2020, pp. 207–212. DOI: 10.1109/ICIEA48937.2020.9248225.
- [29] M. A. Rahman, B. Honan, T. Glanville, P. Hough, and K. Walker, “Using data mining to predict emergency department length of stay greater than 4 hours: Derivation and single-site validation of a decision tree algorithm,” *Emergency Medicine Australasia*, vol. 32, no. 3, pp. 416–421, 2020. DOI: <https://doi.org/10.1111/1742-6723.13421>.